POLITECNICO DI TORINO
Repository ISTITUZIONALE

TTP-free Asymmetric Fingerprinting based on Client Side Embedding

Original
TTP-free Asymmetric Fingerprinting based on Client Side Embedding / Bianchi, Tiziano; Alessandro, Piva. - In: IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. - ISSN 1556-6013. - 9:10(2014), pp. 1557-1568. [10.1109/TIFS.2014.2340581]

Availability:
This version is available at: 11583/2556160 since:

Publisher:
IEEE - INST ELECTRICAL ELECTRONICS ENGINEERS INC

Published
DOI:10.1109/TIFS.2014.2340581

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

18 April 2024

TTP-free Asymmetric Fingerprinting based on Client Side Embedding

T. Bianchi *Member, IEEE,* and A. Piva *Senior Member, IEEE*

***Abstract*—In this paper, we propose a solution for implementing an asymmetric fingerprinting protocol within a client-side embedding distribution framework. The scheme is based on two novel client-side embedding techniques that are able to reliably transmit a binary fingerprint. The first one relies on standard spread-spectrum like client-side embedding, while the second one is based on an innovative client-side informed embedding technique. The proposed techniques enable secure distribution of personalized decryption keys containing the Buyer's fingerprint by means of existing asymmetric protocols, without using a trusted third party. Simulation results show that the fingerprint can be reliably recovered by using either non-blind decoding with standard embedding or blind decoding with informed embedding, and in both cases it is robust with respect to common attacks. To the best of our knowledge, the proposed scheme is the first solution addressing asymmetric fingerprinting within a client-side framework, representing a valid solution to both customer's rights and scalability issues in multimedia content distribution.**

***Index Terms*—Fingerprinting, Buyer-Seller watermarking (BSW) protocol, Client-side embedding, secure watermark embedding.**

I. Introduction

The recent proliferation of various platforms for the distribution of multimedia contents requires the adoption of effective protection measures for preventing copyright violations. Digital watermarking provides a means to embed into the to-be-distributed content a unique code, as a fingerprint, linking the content to a specific recipient.

In the most common case, distribution tracing is made possible by letting the entity selling the content, referred to simply as the Seller, insert a distinct watermark, called a *fingerprint*, identifying the person purchasing the content, referred to as the Buyer, within any copy of data that is distributed. Whenever an unauthorized published content is found, this fingerprint can be used to trace the author of the illegal redistribution [1]–[3].

Most of the existing watermarking techniques for multimedia content protection have been developed to face two important practical issues. One (known in the literature as customer's rights problem) is related to the fact that the distribution server should not know the actual fingerprint embedded into the content, since an accused customer could claim that he/she has been framed by a malicious seller who inserted his/her fingerprint in an arbitrary content. The mere existence of this problem could discredit the forensic tracing architecture and severely limit its adoption. A possible solution to this problem is represented by the asymmetric fingerprinting schemes [4]. In such schemes, only the buyer has access to the fingerprinted content; however, if the seller later finds a copy of the content, the buyer can still be identified and proved guilty in front of a judge. Several asymmetric fingerprinting protocols suitable for realistic multimedia contents, often referred to as Buyer-Seller Watermarking Protocols, exist [5]–[8]: a special class includes those relying only on messages exchanged between the buyer and the seller, without requiring the adoption of a dedicated trusted third party (TTP) [9], [10].

The second issue is related to the system scalability. In a classical distribution model, adopted also by the Buyer-Seller Watermarking Protocols, individually watermarked copies have to be generated and delivered by the distribution server to each user. Since both the computational burden due to watermark embedding and the required bandwidth grow linearly with the number of users, in large-scale systems the server could consume a prohibitive amount of resources. An effective solution to the system scalability problem is provided by client-side embedding [11]. In such schemes, the server distributes the same encrypted copy of the content to all the clients, along with different client-specific decryption keys allowing each user to decrypt a slightly different version of the content, bearing a different watermark. Secure client-side embedding methods suitable for realistic multimedia content have been developed adopting spread-spectrum watermarking [12], informed embedding [13], and vector quantization [14].

Although client-side embedding provides an elegant solution to the system scalability problem, it still suffers of the customer's rights problem, since the server has access to the decryption keys that carry the client-specific watermarks. Some works [15], [16] have proposed to introduce a TTP in order to manage the distribution of the decryption keys, however, again, such a TTP can become quickly overloaded in a realistic system, thus hindering the advantages offered by the client-side embedding. To the best of our knowledge, there is no existing solution that incorporates the aforementioned techniques into an asymmetric fingerprinting protocol, thus solving both the customer's rights problem and the scalability issue.

In this paper, we propose a simple scheme to exploit existing secure asymmetric fingerprinting protocols within a client-side embedding distribution framework.

Namely, we modify the client-side embedding technique proposed in [12] so that it can be used to reliably transmit a binary fingerprint, which enables the secure distribution of

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

T. Bianchi is with the Department of Electronics and Telecommunications, Politecnico di Torino, I-10129, Torino, Italy (e-mail: tiziano.bianchi@polito.it).

A. Piva is with the Department of Information Engineering, University of Florence, 50139, Florence, Italy (e-mail: alessandro.piva@unifi.it).

decryption keys by means of existing TTP-free buyer-seller watermarking protocols.

Thanks to the properties of the used protocol, the server can distribute personalized decryption keys without knowing the actual fingerprint embedded in each key, which eliminates the need of a TTP. At the same time, since the size of a decryption key is much lower than the size of a multimedia content, and a single key can be used for multiple contents, the complexity of running an existing TTP-free buyer-seller protocol, like e.g. that in [9], for the distribution of the keys is still reasonable.

This paper extends a previous work by Bianchi *et al.* [17] under several aspects. An important novelty is the introduction of a client-side informed embedding technique, which enables reliable recovery of the fingerprint by means of blind decoding. We also provide a detailed analysis of the security and the scalability of the proposed technique, as well as new experimental results obtained on an extended dataset and under more realistic conditions.

The paper is organized as follows: In Section II, we briefly review some basic concepts useful to understand the described solutions. In Section III, we introduce the proposed asymmetric protocol, together with two client-side embedding strategies enabling its implementation. In Section IV, we discuss fingerprint decoding for the two described client-side embedding techniques. In Section V, we present some experimental results demonstrating the feasibility of the proposed solutions. Some conclusions will end the paper in Section VI.

II. Preliminaries

In this section, first of all, the notation describing a watermarking process is introduced; next, homomorphic encryption is introduced as building block of the proposed solution. Finally, buyer seller protocols, with particular attention to the ones based on the use of look-up tables (LUT), are defined.

A. Watermarking Model

Given a vector $\mathbf{x} = [x_0, x_1, \ldots, x_{M-1}]$, representing either the original host signal samples or, more generally, a set of features of the host signal, and some to-be-hidden information, represented as a binary vector $\mathbf{b} = [b_0, b_1, \ldots, b_{L-1}]$, an *embedder* inserts the watermark code $\mathbf{b}$ into the host signal to produce a watermarked signal $\mathbf{y}$, usually making use of a secret key sk to control some parameters of the embedding process and allow the watermark recovery only to authorized users. It is often useful to describe the embedding function by introducing a watermark signal $\mathbf{w}$, so that the watermarked signal can be expressed as $\mathbf{y} = \mathbf{x} + \mathbf{w}$.

B. Homomorphic Cryptosystems

A cryptosystem is said to be *homomorphic* with respect to an operation $\star$ if there exists an operator $\phi(\cdot,\cdot)$ such that for any two plain messages m_1 and m_2, we have:

$$\phi(\llbracket m_1 \rrbracket, \llbracket m_2 \rrbracket) = \llbracket m_1 \star m_2 \rrbracket \tag{1}$$

where $\llbracket \cdot \rrbracket$ denotes the encryption operator. Homomorphic encryption allows to perform a set of operations by working on encrypted data. In particular, an additively homomorphic cryptosystem maps an addition in the plaintext domain to an operation in the ciphertext domain, (usually a multiplication). Given two plaintexts m_1 and m_2, the following equalities are then satisfied:

$$\llbracket m_1 \rrbracket \cdot \llbracket m_2 \rrbracket = \llbracket m_1 + m_2 \rrbracket \tag{2}$$

and, as a consequence,

$$\llbracket m \rrbracket^a = \llbracket am \rrbracket \tag{3}$$

where a is a public integer. Additively homomorphic cryptosystems allow then to perform in the encrypted domain additions, subtractions and multiplications with a known (non-encrypted) value (but not division, since it could lead to non integer values), thus providing a way of applying any linear operator in the encrypted domain.

Another desirable property of a homomorphic cryptosystem is the *semantic security*, such that given two encrypted values it is not computationally feasible to decide whether they conceal the same value or not; this property guarantees the confidentiality of the cryptosystem when encrypting data with a restricted set of possible values (for example bits), or when a set of data exhibiting a peculiar correlation structure (for example consecutive signal samples) is encrypted as separate ciphertexts. A well known additively homomorphic and semantically secure asymmetric encryption scheme is the one proposed by Paillier [18].

C. Asymmetric Fingerprinting

In *asymmetric fingerprinting* [4] the Buyer first commits to a secret that only he/she knows (registration phase), then Buyer and Seller follow a protocol (named Buyer-Seller watermarking protocol) after which only the Buyer receives a copy of the watermarked work. However, if the copy is illegally distributed, the Seller can identify the Buyer from whom the copy originated, and prove it to a Judge by using a proper dispute resolution protocol.

A fundamental building block of asymmetric fingerprinting is a functionality that allows Seller and Buyer to jointly perform watermark embedding, in such a way that the original content $\mathbf{x}$ is a private input of the Seller, whereas the fingerprint data $\mathbf{b}$ and thus the watermark $\mathbf{w}$ are a private input of the Buyer. Most recent solutions adopt secure signal processing techniques based on homomorphic encryption, that has been introduced in the previous part of the Section. Let us assume that the Buyer holds a public/private key pair (puk, prk) of an additively homomorphic cryptosystem, like the Paillier one. If Seller and Buyer can share an encryption of the watermark signal $\mathbf{w}$, encrypted with the Buyer's public key puk, then watermark embedding can be performed by the Seller in the encrypted domain as follows

$$\llbracket y_i \rrbracket = \llbracket x_i \rrbracket \cdot \llbracket w_i \rrbracket. \tag{4}$$

where the operation is applied componentwise on the data vector. Indeed, the Seller, knowing the plaintext values of x_i, can compute the ciphertexts $\llbracket x_i \rrbracket$ by relying on the Buyer's public key puk. However, the computed value $\llbracket y_i \rrbracket$ is meaningless

for the Seller, since the private key for decrypting belongs the Buyer, the only one having access to the watermarked content.

D. LUT-based Secure Embedding

In the secure embedding proposed by Celik *et al.* in [12], [19], a distribution server generates a long-term master encryption look-up table $\mathbf{E}$ of size T, whose entries, denoted by $\mathbf{E}(0), \mathbf{E}(1), \ldots, \mathbf{E}(T-1)$, are i.i.d. random variables following a Gaussian distribution $\mathcal{N}(0, \sigma_E)$. The LUT $\mathbf{E}$ will be used to encrypt the content to be distributed to the K_U clients. Next, for the k-th client, the server generates a personalized watermark LUT $\mathbf{W}_k$ whose entries follow a Gaussian distribution $\mathcal{N}(0, \sigma_W)$, and builds a personalized decryption LUT $\mathbf{D}_k$ by combining componentwise the master encryption LUT $\mathbf{E}$ and the watermark LUT $\mathbf{W}_k$:

$$\mathbf{D}_k(t) \quad = \quad -\mathbf{E}(t) + \mathbf{W}_k(t) \tag{5}$$

for $t = 0, 1, \ldots, T-1$. The personalized decryption LUTs are then transmitted once to each client over a secure channel. It is worth noting that the generation of the LUTs is carried out just once at the setup phase.

A content, represented as a vector $\mathbf{x}$ of size *M*, is encrypted by adding to each element of it R entries of the LUT $\mathbf{E}$ pseudo randomly selected according to a content dependent key sk. We assume that each content is linked with a unique key sk, that could be retrieved from a particular content by using for example some robust hashing techniques, as those described in [20], [21].

The obtained encrypted content $\mathbf{c}$ is sent to all the authorized clients along with the key sk. The k-th client can decrypt $\mathbf{c}$ by using his/her personalized decryption LUT $\mathbf{D}_k$, with the final effect that a spread-spectrum watermark sequence is embedded into the decrypted content $\mathbf{y}_k$, through an additive rule (i.e. each content feature is modified according to the rule $y_{k,i} = x_i + w_{k,i}$).

In detail, driven by the content dependent key sk, a set of $M \times R$ values t_{ih} in the range $[0, T-1]$ is generated, where $0 \leq i \leq M-1$, $0 \leq h \leq R-1$. Each of the M content features x_i is encrypted by adding R entries of the encryption LUT identified by the indexes $(t_{i0}, \ldots, t_{i(R-1)})$, obtaining the encrypted feature c_i as follows:

$$c_i \quad = \quad x_i + \sum_{h=0}^{R-1} \mathbf{E}(t_{ih}). \tag{6}$$

Joint decryption and watermarking is accomplished by reconstructing with the content dependent key sk the same sequence of indexes t_{ih} and by adding R entries of the decryption LUT $\mathbf{D}_k$ to each encrypted feature c_i:

$$y_{k,i} = c_i + \sum_{h=0}^{R-1} \mathbf{D}_k(t_{ih}) = x_i + \sum_{h=0}^{R-1} \mathbf{W}_k(t_{ih}) = x_i + w_{k,i} \tag{7}$$

where the i-th watermark component is given as the sum of R entries of the LUT $\mathbf{W}_k$. The result of this operation is the watermarked content $\mathbf{y}_k = \mathbf{x} + \mathbf{w}_k$ identifying the k-th user.

As explained in [12], the parameter R influences the security of the encryption and should be set to $R > 1$ in order to provide resilience against known-plaintext attacks.

III. Asymmetric Client-side Embedding

The key idea of the proposed method is that the decryption LUT in (5) can be alternatively seen as the negative version of the encryption LUT watermarked by a proper signal $\mathbf{W}$ corresponding to the watermarking LUT. Hence, existing buyer-seller watermarking protocols can be used to securely distribute personalized decryption LUTs in such a way that the server does not have access to plaintext versions of those decryption LUTs. However, since existing TTP-free protocols require the buyer to be identified by a unique binary fingerprint, the watermarking LUT must be properly modified so as to embed a binary message into the content and guarantee that the embedded message can be reliably decoded from a possibly modified watermarked content. In the following, we will describe the main tools required to implement the proposed asymmetric version of client-side embedding.

A. Secure Distribution of Personalized Decryption LUTs

Let us assume that the k-th user is identified by the L-bit fingerprint $\mathbf{b}_k$. In the proposed system, the fingerprint is encoded using a binary antipodal modulation, yielding the to be transmitted message $\mathbf{m}_k$, where $m_{k,l} = \sigma_W(2b_{k,l} - 1)$, $0 \leq l \leq L-1$. Hence, the watermarking LUT of the k-th user can be obtained as

$$\mathbf{W}_k = \mathbb{G}\mathbf{m}_k \tag{8}$$

where $\mathbb{G}$ is a $T \times L$ encoding matrix. Namely, $\mathbb{G}$ can be thought as the generator matrix of a linear block code over the set of real numbers [22], [23]. Several choices are possible for $\mathbb{G}$: a really simple solution is to use a repetition code, i.e., $\mathbb{G}$ has only one entry equal to one for each row and approximately T/L entries equal to one for each column. Another solution is to generate the elements of $\mathbb{G}$ as i.i.d. Gaussian variables with zero mean and variance $1/L$.

Since the encoding is linear, the personalized decryption LUT $\mathbf{D}_k$ can be obtained in a secure way by using a simple protocol based on an additively homomorphic cryptosystem. Let us assume that by executing a secure buyer-seller protocol like the one described in [10] the Server obtains an encryption of the Client's fingerprint $[\![\mathbf{b}_k]\!] = [[\![b_{k,0}]\!], [\![b_{k,1}]\!], \ldots, [\![b_{k,L-1}]\!]]$, encrypted with the Client's public key, together with a proper proof of identity. Thanks to the homomorphic properties of the cryptosystem, the Server can compute a rescaled encrypted message as

$$[\![\tilde{m}_{k,l}]\!] = [\![b_{k,l}]\!]^2 [\![1]\!]^{-1} = [\![2b_{k,l} - 1]\!]. \tag{9}$$

In a similar way, each entry of the Client's personalized LUT can be directly computed in the encrypted domain as

$$[\![\mathbf{D}_k(j)]\!] = [\![\mathbf{E}(j)]\!]^{-1} \prod_{l=0}^{L-1} [\![\tilde{m}_{k,l}]\!]^{\sigma_W \mathbb{G}(j,l)}. \tag{10}$$

Finally, the Server can send the encrypted LUT $[\![\mathbf{D}_k]\!]$ to the Client, who decrypts it with his/her private key obtaining

$$\mathbf{D}_k = -\mathbf{E} + \mathbb{G}\mathbf{m}_k. \tag{11}$$

In practice, equation (10) requires that both $\mathbf{E}(j)$ and $\sigma_W \mathbb{G}(j,l)$ are expressed as integer values to be used with

an additively homomorphic cryptosystem defined on modular arithmetic, like Paillier's cryptosystem. This can be achieved by representing such values according to a fixed point representation. For example, when using n_m bits for the magnitude part and n_f bits for the fractional part, the corresponding integer representation of $\mathbf{E}(j)$ can be obtained by computing $\mathrm{round}(\mathbf{E}(j)2^{n_f})$ and clipping the result to $n_m + n_f$ bits. At the Client's side, the correct values of $\mathbf{D}_k$ are obtained by dividing the decrypted integer values by 2^{n_f}. Since a secure homomorphic cryptosystems permits to employ fixed point representations with many bits, the degradation with respect to floating point arithmetic is usually negligible [24].

B. Client-side Standard Embedding

The easiest way of embedding the fingerprint encoded as in (8) in a multimedia content is to directly employ the LUT-based embedding technique described in Section II-D, which in the following will be referred to as client-side standard embedding (CSSE). In order to have a more compact notation, the LUT-based encryption in (6) can be modeled by adding to the signal the product of the encryption LUT $\mathbf{E}$ and a proper binary matrix $\mathbb{T}$ defined according to the sequence of indexes t_{ih}, i.e.,

$$\mathbf{c} = \mathbf{x} + \mathbb{T}\mathbf{E} \tag{12}$$

where $\mathbb{T}$ is a $M \times T$ binary matrix defined as

$$\mathbb{T}(i,j) = \begin{cases} 1 & t_{ih} = j,\ h = 0, \ldots, R-1 \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Hence, CSSE can be obtained in the following way

$$\mathbf{y} = \mathbf{c} + \mathbb{T}\mathbf{D}_k = \mathbf{x} + \mathbb{T}\mathbb{G}\mathbf{m}_k = \mathbf{x} + \tilde{\mathbb{G}}\mathbf{m}_k \tag{14}$$

that is, the fingerprint $\mathbf{m}_k$ is encoded in the watermarked signal by means of the equivalent linear block code defined by the $M \times L$ generator matrix $\tilde{\mathbb{G}} = \mathbb{T}\mathbb{G}$.

C. Client-side Informed Embedding

Informed embedding methods are a class of data hiding schemes where the watermarking problem is viewed as one of communications with side information at the encoder [25]. These systems can achieve host-interference rejection by adequately exploiting in system design knowledge of the host signal at the encoder, in such a way that in the absence of attacks the probability of decoding error is equal to zero. Within this class of methods, Quantization Index Modulation (QIM) [26], using as embedding rule the quantization of some content features, is widely adopted due to its good performance.

A client-side embedding technique relying on quantization-based watermarking has been proposed in [13], exploiting a variation of spread transform dither modulation (STDM). In the following, we will show how similar ideas can be extended also to the proposed approach. According to the QIM principle [26], it is possible to define an informed embedding rule by choosing a set of quantizers, each associated to a different message $\mathbf{m}$, and quantizing $\mathbf{x}$ with the quantizer corresponding to the to-be-transmitted message. In order to apply QIM to the proposed scheme, it is useful to introduce the notion of lattice quantizer. A lattice $\mathcal{L}$ can be informally defined as the set of points in a M-dimensional space obtained as integer linear combinations of the columns of a $M \times L$ lattice generator matrix $\mathbb{L}$, i.e.,

$$\mathcal{L}(\mathbb{L}) = \left\{ \mathbf{r} | \mathbf{r} = \mathbb{L}\mathbf{z}, \mathbf{z} \in \mathbb{Z}^L \right\}. \tag{15}$$

A *lattice quantizer* is then defined as a rule that associates to each vector $\mathbf{x}$ the nearest lattice point in $\mathcal{L}$, i.e.,

$$Q_{\mathcal{L}}(\mathbf{x}) = \arg\min_{\mathbf{r} \in \mathcal{L}} ||\mathbf{x} - \mathbf{r}||_2. \tag{16}$$

The set of all vectors that are quantized to the same lattice point $\mathbf{r}$ is called the *Voronoi cell* associated to $\mathbf{r}$. The volume of the Voronoi cell is often referred to as the lattice volume and can be computed as $V_{\mathcal{L}} = \det(\mathbb{L}^T\mathbb{L})^{1/2}$.

When $M > L$, the matrix $\mathbb{L}$ defines a L-dimensional lattice embedded into a M-dimensional space. Following the STDM approach [13], [26], quantization according to the lattice defined by $\mathbb{L}$ can be viewed as projecting a M-dimensional vector $\mathbf{x}$ onto the subspace spanned by $\mathbb{L}$ and approximating the projection $\mathbf{p}$ by the nearest lattice point. The projection of $\mathbf{x}$ onto the subspace spanned by $\mathbb{L}$ can be expressed as $\mathbf{p} = \mathbb{L}(\mathbb{L}^T\mathbb{L})^{-1}\mathbb{L}^T\mathbf{x}$ and the quantized projection is obtained as

$$\mathbf{p}_Q = Q_{\mathcal{L}(\mathbb{L})}(\mathbf{p}) \tag{17}$$

The final quantized vector can be obtained by substituting the quantized projection for the original projection, i.e.,

$$\mathbf{x}_Q = \mathbf{x} - \mathbf{p} + \mathbf{p}_Q. \tag{18}$$

QIM based on a lattice quantizer can be defined by relying on the concept of lattice partitioning [27]–[29]. Given a lattice $\mathcal{L}$, we can define a sublattice $\mathcal{L}_c$ as a subset of the points in $\mathcal{L}$ that is itself a lattice. Starting from a sublattice $\mathcal{L}_c$ and a point $\mathbf{r} \in \mathcal{L}$, the set of all points obtained as the sum of $\mathbf{r}$ and a point in $\mathcal{L}_c$ is called a *coset* of $\mathcal{L}_c$. It can be shown that given $\mathcal{L}$ and $\mathcal{L}_c$, there are exactly $V_{\mathcal{L}_c}/V_{\mathcal{L}}$ distinct cosets. The set of all the cosets of $\mathcal{L}_c$ with respect to $\mathcal{L}$ constitutes a *lattice partition* of $\mathcal{L}$, i.e., the union of all the possible distinct cosets yields the lattice $\mathcal{L}$. Lattice-QIM is then defined as follows: 1) choose a lattice partition with 2^L cosets; 2) associate each coset $\mathcal{L}_{\mathbf{m}}$ with one of the possible 2^L messages; 3) encode message $\mathbf{m}$ as $\mathbf{y} = Q_{\mathcal{L}_{\mathbf{m}}}(\mathbf{x})$, where $Q_{\mathcal{L}_{\mathbf{m}}}$ denotes the quantizer defined by the translated lattice $\mathcal{L}_{\mathbf{m}}$.

By looking at (14), it is evident that the proposed technique encodes the fingerprint as a point of a translated lattice defined by the generator matrix $\mathbb{L} = 2\sigma_W\tilde{\mathbb{G}}$. Namely, the watermarking signal can be expressed as

$$\mathbf{w}_k = \mathbb{L}\mathbf{b}_k - \mathbf{r}_W \tag{19}$$

where $\mathbf{r}_W = [\sigma_W, \sigma_W, \ldots, \sigma_W]$. Hence, it is possible to define a proper embedding rule by exploiting the lattice generated by $\mathbb{L}$, or, more specifically, its translation by $-\mathbf{r}_W$.

In order to adopt lattice-QIM in the proposed client-side framework, we first choose a proper lattice partition of $\mathcal{L}(\mathbb{L})$. Such a partition will be defined according to the cosets of the sublattice generated by $\mathbb{L}_c = 2\mathbb{L} = 4\sigma_W\tilde{\mathbb{G}}$. It is easy to verify that there are 2^L cosets, each obtained by adding to the lattice

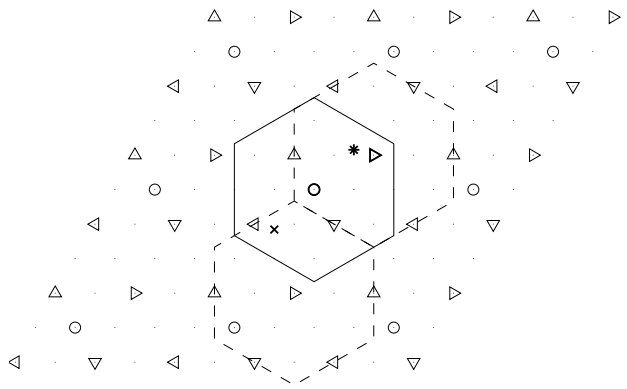

Fig. 1. Example of lattice-QIM performed by CSIE for $L = 2$. The circles represent the points of $\mathcal{L}(\mathbb{L}_c)$. The triangles represent the four possible cosets of $\mathcal{L}(\mathbb{L}_c)$ translated by $-\mathbf{r}_W$. The solid line indicates the Voronoi region of $\mathcal{L}(\mathbb{L}_c)$. The asterisk and the cross represent two possible host signal projections $\mathbf{p}$. Let us assume that we want to encode the fingerprint $\mathbf{b} = [1, 1]$. In both cases, $\mathbf{p}$ is quantized on the same point of $\mathcal{L}(\mathbb{L}_c)$, denoted by the bold circle, and we get the same watermarked projection, denoted by the bold triangle. Since quantization depends only on the host, it can be done at the server's side, enabling client side embedding of the watermark. However, in the "cross" case the watermarked projection does not correspond to the closest coset point, as can be noticed by the Voronoi regions of the coset corresponding to $\mathbf{b} = [1, 1]$, plotted in dashed lines.

generated by $\mathbb{L}_c$ an offset vector $\mathbb{L}\mathbf{b}_k$, where $\mathbf{b}_k$ is one of the possible 2^L binary vectors of length L. Then, each fingerprint is encoded by quantizing the projection $\mathbf{p}$ according to the translation of the corresponding coset by $-\mathbf{r}_W$. It is easy to verify that this strategy encodes each fingerprint as a point obtained by adding to the lattice generated by $\mathbb{L}_c$ an offset vector given by $\mathbf{w}_k = \mathbb{L}\mathbf{b}_k - \mathbf{r}_W$.

In practice, the quantization rule is relaxed by first quantizing the projection $\mathbf{p}$ according to the lattice generated by $\mathbb{L}_c$ and then adding an offset vector $\mathbf{w}_k$, i.e.,

$$\mathbf{p}_Q = Q_{\mathcal{L}(\mathbb{L}_c)}(\mathbf{p}) + \mathbf{w}_k. \tag{20}$$

The above rule does not guarantee to choose the translated coset point that is closest to $\mathbf{p}$. However, the corresponding embedding rule can be implemented as the sum of a content dependent part and a fingerprint dependent part as

$$\mathbf{y} = \mathbf{x} + Q_{\mathcal{L}(\mathbb{L}_c)}(\mathbf{p}) - \mathbf{p} + \mathbf{w}_k = \mathbf{x}_Q + \mathbf{w}_k \tag{21}$$

which enables its use in a client-side setting. We point out that the shift by $-\mathbf{r}_W$ with respect to the cosets of the lattice generated by $\mathbb{L}_c$ is needed in order to have an unbiased error when defining the above embedding rule. A visual example of the proposed embedding technique is shown in Fig. 1.

The corresponding client-side informed embedding (CSIE) can be obtained by defining the encrypted content as

$$\mathbf{c} = \mathbf{x}_Q + \mathbb{T}\mathbf{E} \tag{22}$$

and decrypting it by using the decryption LUT defined in (11), which yields

$$\mathbf{y} = \mathbf{c} + \mathbb{T}\mathbf{D}_k = \mathbf{x}_Q + \tilde{\mathbb{G}}\mathbf{m}_k \tag{23}$$

that is equivalent to the embedding rule in (21).

Since searching for the closest lattice point in an arbitrary lattice is in general a NP-hard problem [30], in our implementation we choose to compute $\mathbf{p}_Q$ by rounding to the nearest integer the coordinates of $(\mathbb{L}_c^T\mathbb{L}_c)^{-1}\mathbb{L}_c^T\mathbf{x}$, i.e.,

$$\mathbf{p}_Q = \mathbb{L}_c \left[(\mathbb{L}_c^T\mathbb{L}_c)^{-1}\mathbb{L}_c^T\mathbf{x}\right] + \mathbf{w}_k. \tag{24}$$

where $[\mathbf{r}]$ denotes rounding each component of $\mathbf{r}$ to the nearest integer. It is worth noting that the above strategy is equivalent to choosing the nearest lattice point if $\mathbb{L}_c$ is orthogonal. Noticeably, the above technique can alternatively be seen as a STDM where each bit of the fingerprint is embedded using a different spreading vector corresponding to a column of $\tilde{\mathbb{G}}$.

IV. Fingerprint Decoding

In this section, we will discuss the decoding of the transmitted fingerprint from the received watermarked content. Since the scheme is asymmetric, the decoder does not know the messages $\mathbf{m}_k$, so it can not employ a correlation detector as in [12]. Instead, the detector obtains an estimated fingerprint $\hat{\mathbf{b}}_k$ and verifies whether it matches with a recorded Client, using the proof of identity provided by the underlying buyer-seller protocol. CCSE and CSIE require different decoders, that will be separately treated.

A. Decoders for Standard Embedding

Let us assume that the watermark decoder receives a copy of a signal watermarked according to (14) and corrupted by an additive noise, i.e., the received signal is

$$\mathbf{y}' = \mathbf{y} + \mathbf{n} = \mathbf{x} + \tilde{\mathbb{G}}\mathbf{m}_k + \mathbf{n}. \tag{25}$$

When the original signal is available at the decoder, its interference can be removed and decoding can be performed on the signal $\mathbf{y}'' = \mathbf{y}' - \mathbf{x} = \tilde{\mathbb{G}}\mathbf{m}_k + \mathbf{n}$. Otherwise, blind decoding can be obtained by directly using the received signal $\mathbf{y}'$ and considering $\mathbf{x}$ as an additional noise term.

Several decoding strategies can be considered to recover the Client's fingerprint $\hat{\mathbf{b}}_k$. When the signal is corrupted by additive white Gaussian noise (AWGN), the maximum likelihood decoder is the *Minimum Distance (MD)* decoder

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{\arg\min_{\mathbf{m}} ||\mathbf{y}'' - \tilde{\mathbb{G}}\mathbf{m}||_2\right\}. \tag{26}$$

where we define

$$\text{sgn}\{a\} = \begin{cases} 1 & a > 0 \\ 0 & a \leq 0. \end{cases} \tag{27}$$

The MD decoder achieves optimal decoding performance when all the fingerprints are equiprobable. Nevertheless, the direct implementation of the MD decoder is a NP-hard problem, since it requires the enumeration of all the possible fingerprints. A practical solution is to use sphere decoding, which with high probability achieves the MD solution in polynomial time.

Alternatively, suboptimal decoders can be used. In this paper, we will consider the *Matched Filter (MF)* decoder

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{\tilde{\mathbb{G}}^T\mathbf{y}''\right\} \tag{28}$$

and the *Pseudo-Inverse (PI)* decoder

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{(\tilde{\mathbb{G}}^T\tilde{\mathbb{G}})^{-1}\tilde{\mathbb{G}}^T\mathbf{y}''\right\}. \quad (29)$$

MF and PI decoders are based on standard suboptimal receiver commonly adopted in digital communications. Namely, the PI decoder corresponds to zero-forcing equalization followed by hard decision.

B. Decoders for Informed Embedding

The decoding of a fingerprint embedded according to (23) can be performed by looking for the translated coset which is closest to the received signal $\mathbf{y}'$. In this case, the decoding is always blind, since the original signal $\mathbf{x}$ is not required for deciding the closest coset.

In general, the corresponding MD decoder can be obtained as

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{\arg\min_{\mathbf{m}}\min_{\mathbf{r}\in\mathbb{Z}^L}||\mathbf{y}' - 4\sigma_W\tilde{\mathbb{G}}\mathbf{r} - \tilde{\mathbb{G}}\mathbf{m}||_2\right\}. \quad (30)$$

Alternatively, we note that by quantizing $\mathbf{y}'$ to the nearest point of the fine lattice generated by $\sigma_W\tilde{\mathbb{G}}$, each component of a translated coset is represented by the integer coordinates $4z\pm1$, where $z\in\mathbb{Z}$. Hence, MD decoding can be equivalently achieved as

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{\arg\min_{\mathbf{r}\in\mathbb{Z}^L}||\mathbf{y}' - \sigma_W\tilde{\mathbb{G}}\mathbf{r}||_2 \mod 4\right\} \quad (31)$$

where the remainder of the division by 4 is computed independently for each component and mapped in the interval $[-2, 2)$.

The above decoder is optimal for an AWGN channel, however it requires sphere decoding which may be to expensive in practical situations. Suboptimal decoders can be obtained by approximating the quantization of $\mathbf{y}'$ according to the lattice generated by $\sigma_W\tilde{\mathbb{G}}$ as in (24), leading to the following PI decoder

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{\left[\frac{1}{\sigma_W}\left(\tilde{\mathbb{G}}^T\tilde{\mathbb{G}}\right)^{-1}\tilde{\mathbb{G}}^T\mathbf{y}'\right] \mod 4\right\}. \quad (32)$$

Under the assumption that the lattice is near orthogonal, i.e., $\tilde{\mathbb{G}}^T\tilde{\mathbb{G}} \approx \frac{RT}{L}\mathbb{I}_L$, we can also approximate the PI decoder using the following scaled MF decoder

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{\left[\frac{L}{RT\sigma_W}\tilde{\mathbb{G}}^T\mathbf{y}'\right] \mod 4\right\}. \quad (33)$$

V. Performance Evaluation

A. Security

The security analysis of the proposed scheme depends on several aspects, including the security of the encrypted content $\mathbf{c}$, the ability to track dishonest Clients and the impossibility of framing honest Clients.

As to the encrypted content, if the power of the encryption LUT is much larger than the signal power, the encrypted signal $\mathbf{c}$ is virtually indistinguishable form random noise. Security can be further enhanced by computing $\mathbf{c}$ modulo an integer Z: if $Z \ll \sigma_E$, then the values of the encrypted signal are approximately uniformly distributed on $[0, Z-1]$, irrespective of the signal values. A dishonest Client could try to exploit the knowledge of the sequence of indexes t_{ih} for estimating the content from $\mathbf{c}$. In [12], it is shown that such an attack has a very low performance as long as the LUT length T is of the same order as the content length M.

The ability to track dishonest Clients is directly inherited from the underlying Buyer-Seller protocol used to distribute decryption LUTs. If a Server can successfully decode a Client's fingerprint from an illegally redistributed copy, he can link the copy to the Client through the proof of identity provided by the protocol. In order to prevent a dishonest Client from removing his watermark from the decryption LUT, the generator matrix $\mathbb{G}$ can be kept secret, acting as a secret watermarking key. Moreover, the Client's fingerprint can be easily randomized by the Server in the encrypted domain [9], so that neither the Client nor the Server knows the actual fingerprint embedded into the decryption LUT. The Client could also try to remove the watermark from the decrypted content: in the next section, we will see that such an attack is usually unsuccessful, unless the content is severely degraded.

A coalition of dishonest Clients could compare their respective decryption LUTs in order to remove the watermark, the so called *collusion* attack. An effective way to withstand collusion attacks is to use an anticollusion code in the design of the fingerprint [31]. In the proposed scheme, anticollusion codes could be used directly in the underlying Buyer-Seller protocol, for example by using the solution in [32], which guarantees that the fingerprint $\mathbf{b}_k$ obtained at the end of the protocol is a Tardos' code [33], [34]. An alternative way is to assign different generator matrices to different Clients. If matrices $\mathbb{G}$ are independently generated, codes generated by different $\mathbb{G}$s will be nearly orthogonal, achieving similar anticollusion performance as the orthogonal codes proposed in [35].

Finally, the asymmetry of the underlying BS protocol guarantees that an innocent Client can not be framed by a dishonest Server, since the Server only sees encrypted versions of the Client's fingerprint and decryption LUT.

B. Scalability

The proposed solution requires the Server to compute and distribute encrypted decryption LUTs. The computational complexity of the proposed scheme is therefore bigger than the complexity of the plaintext solution in [12]. Nevertheless, the solution in [12] does not protect customer's rights, unless it is complemented with a dedicated TTP. Moreover, by using a composite representation of the LUTs [36] the bandwidth required to transmit an encrypted LUT increases only according to the cryptosystem expansion factor, i.e., by a factor two with Paillier, or even less using Paillier extensions [37].

Currently, the only solutions protecting customer's rights without a TTP are based on Server's side encrypted domain embedding. With respect to those solutions, the proposed approach offers both computational and communication complexity advantages. In the proposed scheme the Server has to distribute a different encrypted LUT to each client; let us assume that the computational cost for processing a LUT of size T is proportional to T. If those LUTs are used to access N_C contents, then the computational cost per user and

per distributed content will be proportional to T/N_C. In the Server's side scheme, the Server has to distribute different encrypted contents for each client. If we again assume that the computational cost for processing a content of size M is proportional to M, then the computational cost per user and per distributed content is also proportional to M. Hence, the proposed scheme has a computational gain of MN_C/T with respect to a Server's side solution. According to [12], an attacker observing an overall content of size MN_C protected with a LUT of size T, can obtain a fingerprint-free copy with a signal-to-noise ratio (SNR) equal to MN_C/T. Hence, the computational gain of the proposed solution is equal to the SNR attainable by a possible attacker exploiting the LUT structure. Namely, if SNR = 20 dB, which corresponds to a poor quality content, the proposed scheme is 100 times less complex than a Server's side scheme.

As to communication complexity, if there are N_U clients, in the proposed scheme the Server distributes N_U LUTs of size T plus N_C encrypted contents of size M. Conversely, in a Server's side scheme the Server should distribute $N_U N_C$ encrypted contents of size M. Hence, the proposed scheme uses a fraction $T/(MN_C) + 1/N_U$ of the bandwidth of the Server's side scheme. If $N_U \gg 1$, the bandwidth reduction roughly corresponds to the computational gain.

C. Experimental Results

For the experimental validation of the proposed technique, we have simulated a system performing client-side embedding on digital images. We have considered a dataset of 100 gray scale uncompressed 8 bit images, each having resolution 1024×1024 pixels. The images represent a variety of subjects, including people, landscapes, building, close objects. For each image, the signal $\mathbf{x}$ has been obtained by applying a 8×8 discrete cosine transform (DCT) to the image and taking 4 DCT coefficients for each 8×8 block, corresponding to the coefficients between the 7th and 10th positions according to the zig-zag ordering used by JPEG standard. In order to avoid interference with JPEG compression, the DCT 8×8 grid has been shifted by four pixels in both vertical and horizontal directions. This resulted in a vector $\mathbf{x}$ of 2^{16} components.

Each image has been encrypted by using an encryption LUT $\mathbf{E}$ with power $\sigma_E^2 = 10^6$. After adding the elements of $\mathbf{E}$ to the selected DCT coefficients, the images have been reconstructed by using an inverse block DCT and pixel values have been mapped to 9 bit values by applying rounding and a modulo 512 operation. An analogous sequence of operations have been performed when decrypting the images with the decryption LUT $\mathbf{D}$. The use of the modulo operation guarantees that the encryption is perfectly reversible as long as $\mathbf{D} = -\mathbf{E}$. When $\mathbf{D} = -\mathbf{E} + \mathbf{W}$, some pixels may exceed the range $[0, 255]$ in the watermarked image: the use of 9 bits guarantees that those pixel values can be detected after decryption and clipped to the range $[0, 255]$. Both $\mathbf{E}$ and $\mathbf{D}$ have been encoded using a fixed point representation with $n_m = 13$ bits for the magnitude part and $n_f = 18$ bits for the fractional part, so as to allow for an encrypted domain implementation using modular arithmetic as explained at the end of Section III-A.

In all experiments, the LUT size has been set to $T = 2^{16}$ and $R = 4$ LUT entries are added together to encrypt each element. We simulated the embedding and subsequent decoding of a 128 binary fingerprint. Two encoding strategies were considered, repetition coding (RC) and i.i.d. random Gaussian coding (GC), as described in Section III. As to decoding, we considered MF and PI decoding, both in the nonblind and in the blind case. For each image, 100 independent tests were performed by randomly generating different encryption LUTs, different fingerprints, and different encoding matrices $\mathbb{G}$. This yielded $100 \times 100 = 10000$ transmitted fingerprints for each experiment.

The decoding performance has been evaluated by measuring the fingerprint error rate (FER), corresponding to the ratio of erroneously decoded fingerprints to the overall transmitted fingerprints. In the simulations, a fingerprint is correctly decoded when all the fingerprint bits are correctly detected. We remark that this is a worst case scenario for fingerprint detection, since it corresponds to the lowest probability of falsely accusing an innocent user. Namely, if we assume that different fingerprints are independently drawn, with this decoding convention the probability of falsely accusing an innocent user can be upper bounded as $P_{fa} \leq N_U 2^{-L}$, where N_U is the number of users. For $L = 128$, this probability is virtually zero even if $N_U \approx 10^9$, which is a very large number of users for any possible application. In order to evaluate the quality of the watermarked image with respect to the original image, we also measured the peak signal-to-noise ration (PSNR) and the mean structural similarity (MSSIM) index [38].

1) Performance of CSSE: A first set of experiments considered the decoding performance in the absence of attacks, by testing different watermarking powers corresponding to values $\sigma_W^2 \in [10^{-7}, 10^3]$. The decoding performance in the nonblind case can be observed in Fig. 2. The results show that for $\sigma_W^2 > 10^{-2}$ all the strategies achieve a good fingerprint decoding performance, with GC being slightly better than RC and PI decoding being slightly better than MF decoding. Namely, for $\sigma_W^2 = 10^{-2}$ GC using PI decoding is able to correctly decode more than 95% of the fingerprints.

Fig. 3 shows the decoding performance in the blind case. As expected, the performance is significatively worse than in the nonblind case. Namely, all strategies, except RC using MF decoding, can correctly decode more than 90% of the fingerprints only when $\sigma_W^2 \geq 10$.

The effect of the watermarking strength on the reconstructed images can be appreciated from Fig. 4 and Fig. 5, showing the obtained values of PSNR and MSSIM index, respectively. For each encoding strategy and each value of σ_W^2, we show the average, minimum, and maximum values obtained over the whole dataset. The results show that a good quality of the watermarked image can be achieved as long as $\sigma_W^2 \leq 1$, whereas the quality sharply degrades for $\sigma_W^2 > 10$. This means that when using nonblind decoding the proposed method can achieve very good decoding performance without significantly affecting the quality of the watermarked image. As a visual example, a watermarked image at $\sigma_W^2 = 1$, using GC, is shown in Fig. 17-(b).

A second set of experiments were conducted in the presence

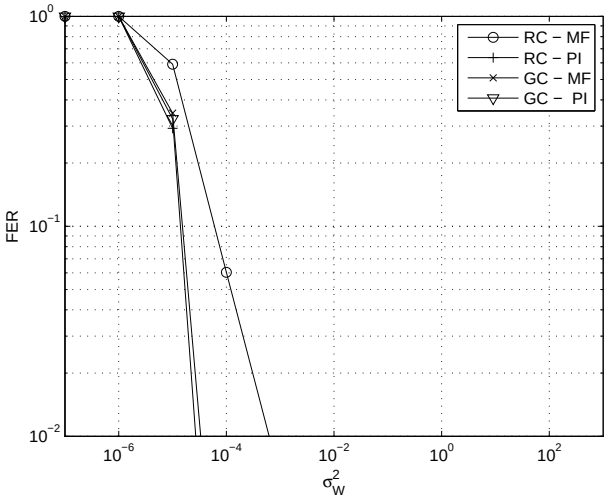

Fig. 2. FER performance of CSSE in the absence of attacks for different encoding and decoding strategies, considering nonblind decoding.

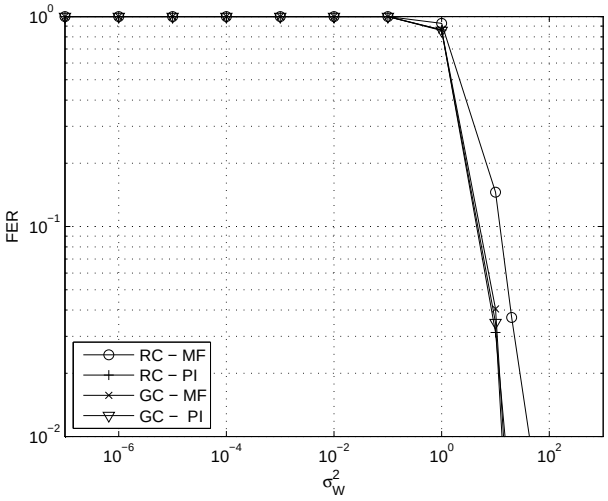

Fig. 3. FER performance of CSSE in the absence of attacks for different encoding and decoding strategies, considering blind decoding.

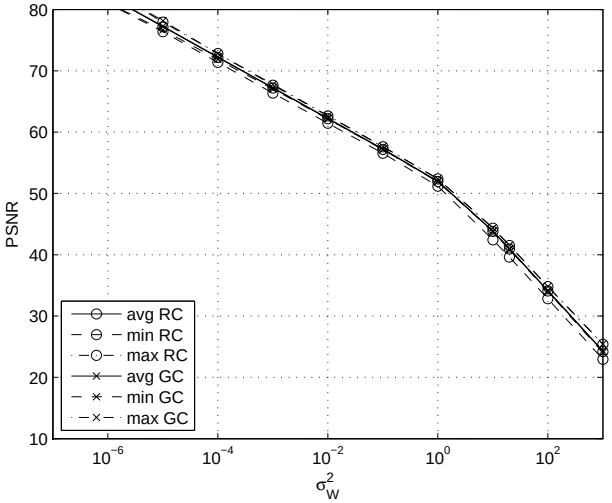

Fig. 4. PSNR obtained by CSSE in the absence of attacks for different encoding strategies.

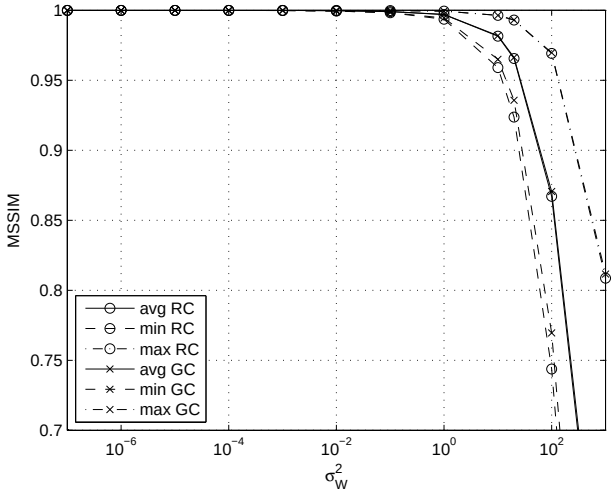

Fig. 5. MSSIM obtained by CSSE in the absence of attacks for different encoding strategies.

of attacks, for $\sigma_W^2 = 1$. In this case, we considered only nonblind decoding, since blind decoding did not provide satisfactory performance in the absence of attacks for the selected watermarking power. The watermarked images were either corrupted by AWGN or compressed using the JPEG standard. In the first case, we considered watermark-to-noise ratios (WNRs) in the range $[-20, 0]$, where we define $\mathrm{WNR} = 10\log_{10}\frac{R\sigma_W^2}{\sigma_N^2}$, being σ_N^2 the variance of the additive noise. In the second case, we considered different JPEG quality factors (QF), ranging from 10 to 100.

Fig. 6 shows the performance of nonblind decoding in the presence of AWGN attack. With the exception of RC using MF decoding, all the strategies achieve very similar performance and guarantee almost error-free decoding of the fingerprint for $\mathrm{WNR} > -15$, which demonstrates a great robustness in the presence of AWGN. Indeed, from the results Fig. 7 and Fig. 8, showing the values of PSNR and MSSIM index after the AWGN attack, it is evident that for $\mathrm{WNR} \leq -15$ the image is so degraded as to be of no practical value.

Fig. 9 shows the performance of nonblind decoding in the presence of JPEG attack. Similarly to the AWGN case, all the strategies achieve very similar performance except RC using MF decoding. In general, the proposed scheme can withstand JPEG compression with a quality factor as low as 40 without showing significant decoding errors and can still correctly decode about 20% of the fingerprints for a quality factor equal to 20. As shown in Fig. 10 and Fig. 11 by the values of PSNR and MSSIM index after the JPEG attack, some of the images may still have an acceptable quality for a quality factor equal to 40, however most of the images have to be largely degraded in order to impede the correct decoding of the fingerprint.

2) Performance of CSIE: Similar sets of experiments were conducted considering the client-side informed embedding strategy described in Section III-C. In this case, only blind decoding has been considered. The decoding performance in the absence of attacks is shown in Fig. 12. The results indicate that the PI decoder achieves a performance similar to that

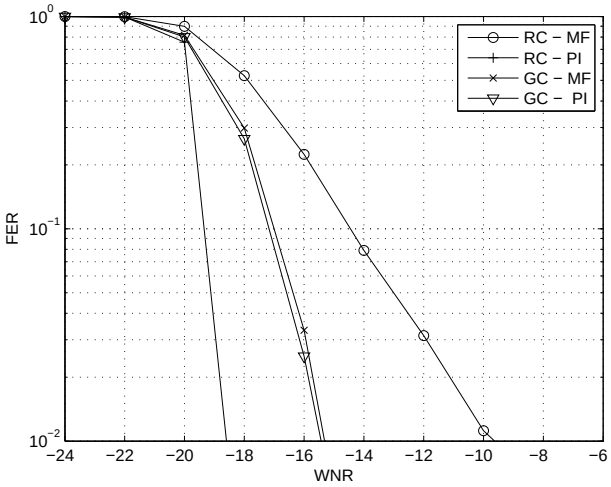

Fig. 6. FER performance of CSSE in the presence of AWGN attack for different encoding and decoding strategies, considering nonblind decoding.

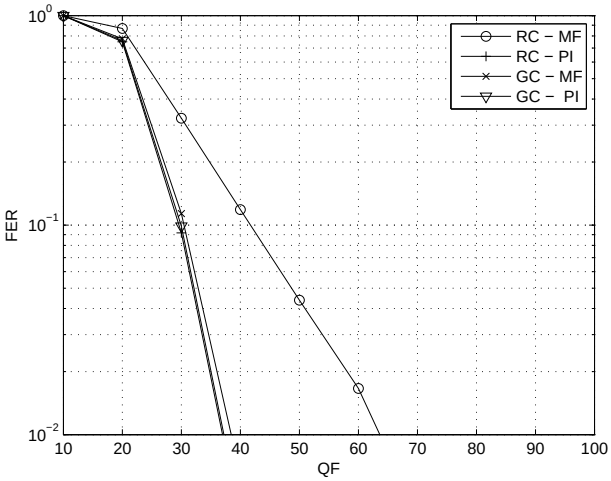

Fig. 9. FER performance of CSSE in the presence of JPEG attack for different encoding and decoding strategies, considering nonblind decoding.

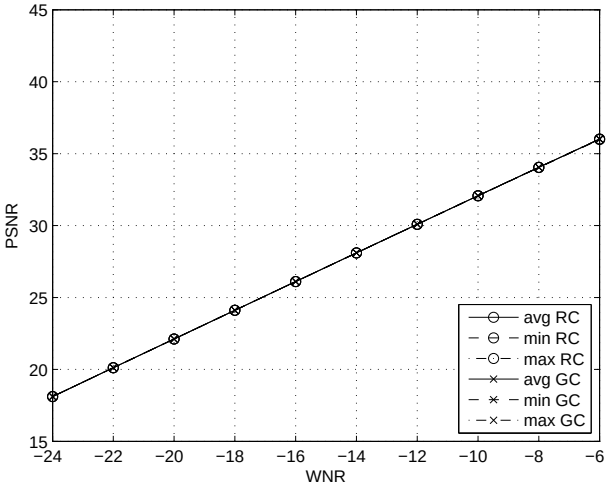

Fig. 7. PSNR obtained by CSSE after AWGN attack for different encoding strategies.

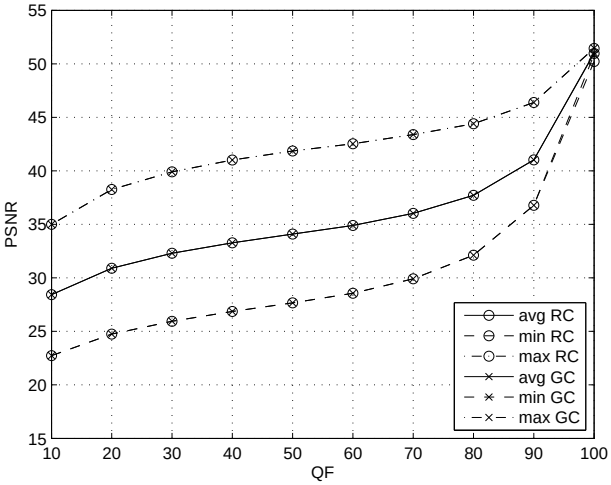

Fig. 10. PSNR obtained by CSSE after JPEG attack for different encoding strategies.

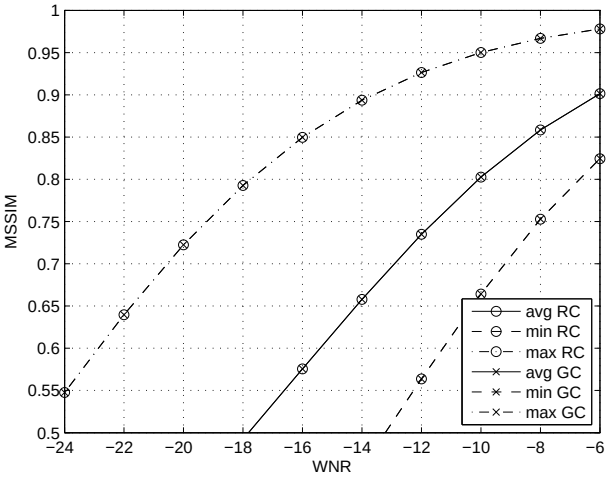

Fig. 8. MSSIM obtained by CSSE after AWGN attack for different encoding strategies.

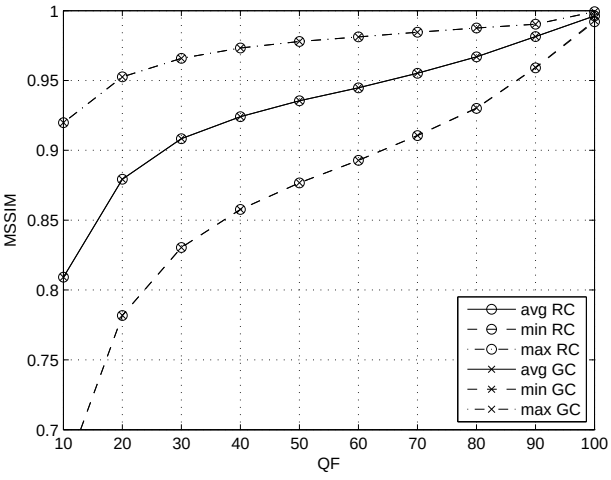

Fig. 11. MSSIM obtained by CSSE after JPEG attack for different encoding strategies.

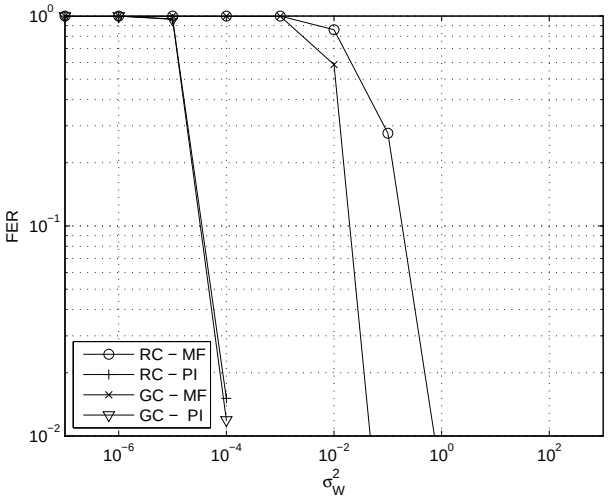

Fig. 12. FER performance of CSIE in the absence of attacks for different encoding and decoding strategies.

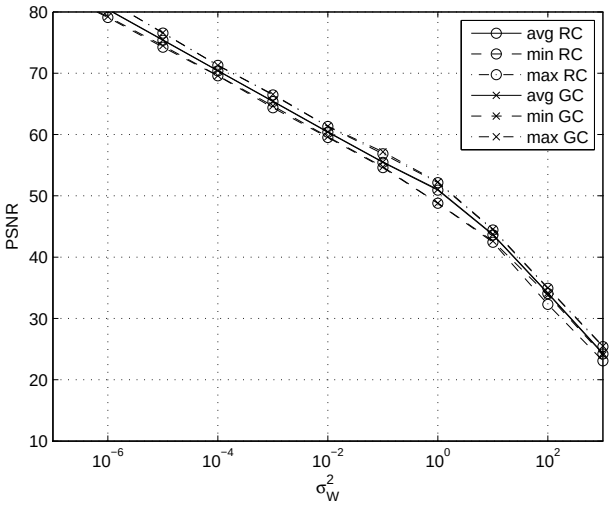

Fig. 13. PSNR obtained by CSIE in the absence of attacks for different encoding strategies.

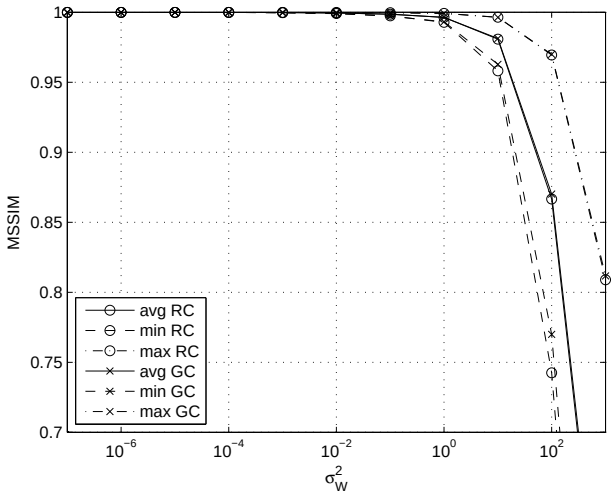

Fig. 14. MSSIM obtained by CSIE in the absence of attacks for different encoding strategies.

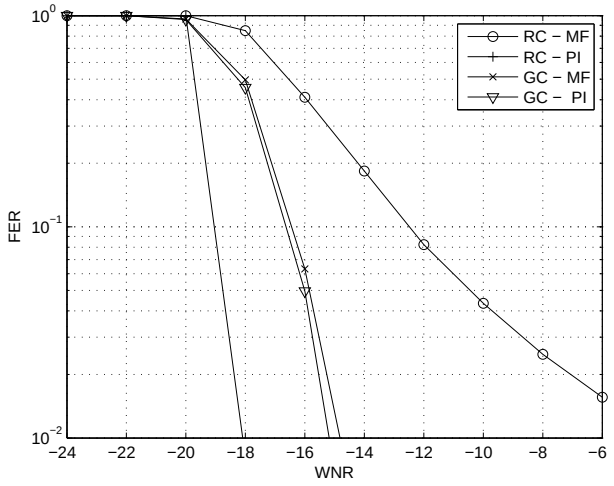

Fig. 15. FER performance of CSIE in the presence of AWGN attack for different encoding and decoding strategies.

of the original client-side embedding system using nonblind decoding, whereas the MF decoder appears slightly impaired in this setting.

The quality of the reconstructed images using the client-side informed embedding technique is very similar to that obtained with the original client-side embedding, as can be seen from Fig. 13 and Fig. 14, showing the obtained values of PSNR and MSSIM index, respectively. As a visual example, a watermarked image at $\sigma^2_W = 1$, using GC, is shown in Fig. 17-(c).

The performance in the presence of AWGN attack, for $\sigma^2_W = 1$, can be observed in Fig. 15. Also in this case the performance of informed embedding is very similar to the performance of the original system. A different behavior can be observed in Fig. 16, showing the decoding performance in the presence of JPEG attack, for $\sigma^2_W = 1$. In this case, client-side informed embedding is less robust than the original system using nonblind decoding, since all decoders show significant decoding errors for a quality below 70. However, it is worth noting that most decoders can still correctly decode about 15% of the fingerprints for a quality factor equal to 30, which can still provide some deterrence effect. Moreover, we point out that a correct fingerprint detection means that all fingerprint bits are correctly detected, which is the most restrictive scenario.

Since in this scenario the distortion on the content is mainly due to the attack, the values of PSNR and MSSIM index obtained after the AWGN attack and the JPEG attack are virtually identical to those obtained using the standard technique (see Fig. 7-8 and Fig. 10-11) and, for the sake of conciseness, they are not reported here.

VI. Conclusions

In this work, a new client-side embedding technique enabling the distribution of multimedia content through an asymmetric fingerprint protocol has been presented. The core idea we have followed is that existing asymmetric protocols,

(a) (b) (c)

Fig. 17. Examples of watermarked images at $\sigma_W^2 = 1$, using GC: (a) original image; (b) CSSE; (c) CSIE. The images show a 600×600 portion of an image in our dataset. Both watermarked images are visually indistinguishable from the original image.

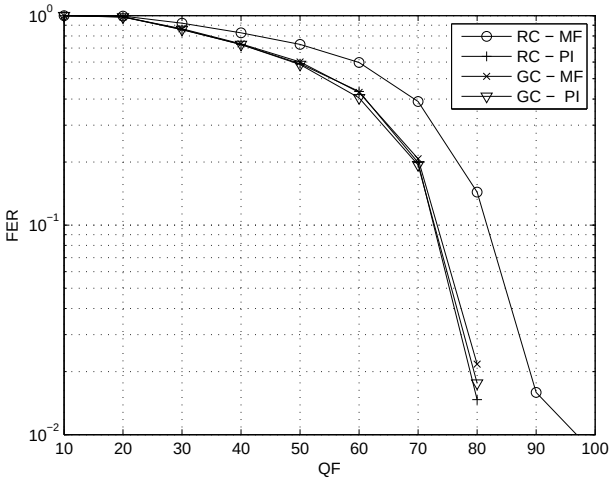

Fig. 16. FER performance of CSIE in the presence of JPEG attack for different encoding and decoding strategies.

not requiring a dedicated trusted third party, can be exploited to securely exchange the personalized decryption keys needed by the client-side embedding scheme. Since the size of a decryption key is much lower than the size of the content to be distributed, and a single key can be used for multiple contents, the proposed solution offers significant advantages with respect to a traditional server-side asymmetric protocol.

In order to make the proposed approach feasible, the Buyer's binary fingerprint has been encoded in the personalized decryption key via linear block coding, which can be securely implemented at the Seller's side by using homomorphic encryption.

Concerning the client-side watermark embedding procedure, we have designed a standard embedding version (CSSE), as well as an informed embedding implementation (CSIE). Simulation results show that the embedded fingerprint can be reliably decoded in both cases from the watermarked content, even when using low watermarking power and in the presence of common attacks, like additive Gaussian noise and JPEG compression. In particular, the CSSE with a non blind Pseudo-Inverse (PI) decoder achieved the best results, followed by the CSIE with PI decoder, that has worse results with respect to the previous version just in case of JPEG attack.

Finally, we believe the proposed scheme can offer a valid solution in multimedia content distribution, since it is able to protect both seller's and customer's rights, and, at the same time, it effectively solves scalability issues.

REFERENCES

[1] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*. Marcel Dekker, 2004.

[2] W. Lin, H. Zhao, and K. Liu, "Game-theoretic strategies and equilibriums in multimedia fingerprinting social networks," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 191 –205, Apr. 2011.

[3] T. Bianchi and A. Piva, "Secure watermarking for multimedia content protection: A review of its benefits and open issues," *IEEE Signal Process. Mag.*, vol. 30, no. 2, pp. 87–96, 2013.

[4] B. Pfitzmann and M. Schunter, "Asymmetric fingerprinting," in *Adv. in Cryptology - EUROCRYPT'96*, ser. LNCS 1070, 1996, pp. 84–95.

[5] N. Memon and P. Wong, "A buyer-seller watermarking protocol," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 643–649, Apr. 2001.

[6] M. Kuribayashi and H. Tanaka, "Fingerprinting protocol for images based on additive homomorphic property," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2129–2139, Dec. 2005.

[7] J. P. Prins, Z. Erkin, and R. L. Lagendijk, "Anonymous fingerprinting with robust QIM watermarking techniques," *EURASIP Journal on Information Security*, vol. 2007, Article ID 31340, 13 pages, 2007.

[8] M. Kuribayashi, "On the implementation of spread spectrum fingerprinting in asymmetric cryptographic protocol," *EURASIP Journal on Information Security*, vol. 2010, pp. 1:1–1:11, Jan. 2010.

[9] M. Deng, T. Bianchi, A. Piva, and B. Preneel, "An efficient buyer-seller watermarking protocol based on composite signal representation," in *Proceedings of the 11th ACM workshop on Multimedia and security*. Princeton, New Jersey, USA: ACM New York, NY, USA, 2009, pp. 9–18.

[10] A. Rial, M. Deng, T. Bianchi, A. Piva, and B. Preneel, "A provably secure anonymous buyer-seller watermarking protocol," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 920 –931, Dec. 2010.

[11] R. J. Anderson and C. Manifavas, "Chameleon—a new kind of stream cipher," in *Proceedings of the 4th International Workshop on Fast Software Encryption — FSE'97*. London, UK: Springer-Verlag, 1997, pp. 107–113.

[12] M. Celik, A. Lemma, S. Katzenbeisser, and M. van der Veen, "Look-up table based secure client-side embedding for spread-spectrum watermarks," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 475–487, 2008.

[13] A. Piva, T. Bianchi, and A. De Rosa, "Secure client-side ST-DM watermark embedding," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 13 –26, Mar. 2010.

[14] C.-Y. Lin, P. Prangjarote, L.-W. Kang, W.-L. Huang, and T.-H. Chen, "Joint fingerprinting and decryption with noise-resistant for vector quantization images," *Signal Processing*, vol. 92, no. 9, pp. 2159 – 2171, 2012.

[15] S. Katzenbeisser, A. Lemma, M. U. Celik, M. van der Veen, and M. Maas, "A buyer–seller watermarking protocol based on secure embedding," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 783–786, Dec. 2008.

[16] G. Poh and K. Martin, "An efficient buyer-seller watermarking protocol based on chameleon encryption," in *Digital Watermarking*, ser. Lecture Notes in Computer Science, H.-J. Kim, S. Katzenbeisser, and A. Ho, Eds. Springer Berlin / Heidelberg, 2009, vol. 5450, pp. 433–447.

[17] T. Bianchi and A. Piva, "TTP-free asymmetric fingerprinting protocol based on client side embedding," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2014)*, Firenze, Italy, 2014, pp. 3987–3991.

[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT 1999*, ser. Lecture Notes in Computer Science, J. Stern, Ed., no. 1592. Springer Verlag, 1999, pp. 223–238.

[19] M. Celik, A. Lemma, S. Katzenbeisser, and M. van der Veen, "Secure embedding of spread-spectrum watermarks using look-up tables," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'07)*, vol. 2, Honolulu, HI, USA, 2007, pp. II–153–II–156.

[20] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *Proc. IEEE Int. Conf. Image Processing (ICIP'00)*, vol. 3, Vancouver, BC, Canada, Sept. 2000, pp. 664–666.

[21] A. Swaminathan, M. Yinian, and M. Wu, "Robust and secure image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 215–230, June 2006.

[22] J. Marshall, T., "Coding of real-number sequences for error correction: A digital signal processing problem," *IEEE J. Sel. Areas Commun.*, vol. 2, no. 2, pp. 381–392, 1984.

[23] Z. Wang and G. Giannakis, "Complex-field coding for OFDM over fading wireless channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 707–720, 2003.

[24] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 86–97, March 2009.

[25] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1127–1141, July 1999.

[26] B. Chen and G. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[27] P. Moulin and R. Koetter, "Data-hiding codes," *Proceedings of the IEEE*, vol. 93, no. 12, pp. 2083–2126, 2005.

[28] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1250–1276, 2002.

[29] L. Pérez-Freire and F. Pérez-González, "Security of lattice-based data hiding against the watermarked-only attack," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 593–610, 2008.

[30] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.

[31] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1897–1905, Sep 1998.

[32] A. Charpentier, C. Fontaine, T. Furon, and I. Cox, "An asymmetric fingerprinting scheme based on Tardos codes," in *Proceedings of the 13th international conference on Information Hiding*, ser. IH'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 43–58.

[33] G. Tardos, "Optimal probabilistic fingerprint codes," *J. ACM*, vol. 55, no. 2, pp. 10:1–10:24, May 2008.

[34] S. Katzenbeisser, B. Škorić, M. Celik, and A.-R. Sadeghi, "Combining Tardos fingerprinting codes and fingercasting," in *Information Hiding*, ser. Lecture Notes in Computer Science, T. Furon, F. Cayre, G. Doërr, and P. Bas, Eds. Springer Berlin / Heidelberg, 2007, vol. 4567, pp. 294–310.

[35] Z. Wang, M. Wu, H. Zhao, W. Trappe, and K. Liu, "Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation," *IEEE Trans. Signal Process.*, vol. 14, no. 6, pp. 804–821, June 2005.

[36] T. Bianchi, A. Piva, and M. Barni, "Composite signal representation for fast and storage-efficient processing of encrypted signals," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 180–187, Mar. 2010.

[37] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of Paillier's probabilistic public-key system," in *4th International Workshop on Practice and Theory in Public-Key Cryptography*, ser. LNCS 1992, 2001, pp. 119–136.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

Tiziano Bianchi (S'03-M'05) received the M.Sc. degree (Laurea) in electronic engineering and the Ph.D. degree in information and telecommunication engineering from the University of Florence, Italy, in 2001 and 2005, respectively.

Since December 2012, he is with the Department of Electronics and Telecommunications, Politecnico di Torino as an Assistant Professor. From 2005 to 2012, he has been with the Department of Electronics and Telecommunications, University of Florence as a Research Assistant. His research interests have involved signal processing in communications and processing of SAR images. Current research topics include multimedia security technologies, signal processing in the encrypted domain, and security aspects of compressed sensing. He has published more than 90 papers on international journals and conference proceedings.

Alessandro Piva (M'04-SM'10) received his Ph.D. degree in Computer Science and Telecommunications Engineering from the University of Florence on 1999.

From 2002 until 2004 he was Research Scientist at the National Inter-university Consortium for Telecommunications (CNIT). Since 2005 he is Assistant Professor at the University of Florence. His research interests lie in the areas of Information Forensics and Security, and of Image and Video Processing. In the above research topics he has been co-author of more than 35 papers published in international journals and 100 papers published in international conference proceedings. He holds 3 Italian patents and an International one on watermarking.

He was IEEE Information Forensics and Security Technical Committee Member; he serves as Associate Editor of the IEEE Transactions on Multimedia, of the EURASIP Journal on Information Security and of the LNCS Transactions on Data Hiding and Multimedia Security, and he served as AE of the IEEE Transactions on Information Forensics and Security and of the IEEE Transactions on Circuits and Systems for Video Technology. He was Technical Co-Chair of IEEE MMSP2004 and of IH&MMSec2014, Publications Chair of IEEE WIFS 2013, and Publications Co-Chair of IEEE ICASSP2014. He also was Co-Organizer of the First IEEE SPS Italy Chapter Summer School on Signal Processing, held in Livorno (Italy), on September 2013, and co-Organizer of the First IEEE IFS-TC Image Forensics Challenge.